

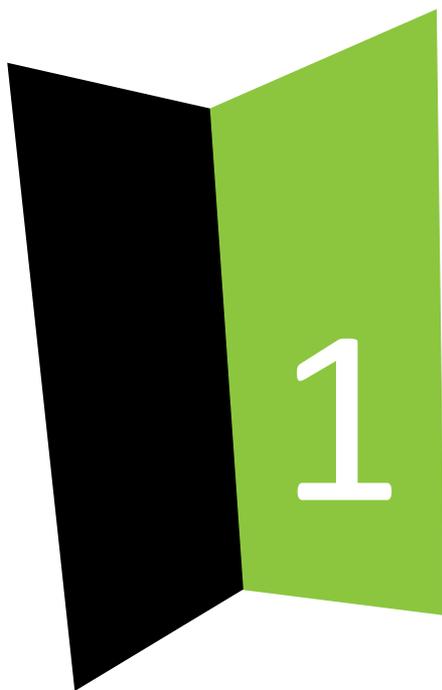
embarcadero[®]

Developer Camp

【C1】 Delphi/C++テクニカルセッション

ゲームUIから考える これからのエンタープライズUX

エークリッパー・インク
羽生章洋



アジェンダ

アジェンダ

- エンタープライズモバイルのポテンシャルと課題
- あらためてUIということを考える
- FireMonkeyでのアプローチ



自己紹介

自己紹介

羽生 章洋 (HABU Akihiro)

エークリッパー・インク
代表

1968年生まれ。大阪育ち。

1989年、桃山学院大学社会学部社会学科を中退後、
2つのソフトウェア会社にて様々な業種・業態向けシステム開発に携わる。
この頃Delphiと出会い、当時のボーランド大阪支店にてセミナー講師も務める。

その後、アーサーアンダーセン・ビジネスコンサルティングに所属。
ERPコンサルタントとして企業改革の現場に従事。
更にその後、 트레이ダーズ証券株式会社の新規創業にIT部門ディレクターとして、
さらにマネースクウェアジャパン株式会社の新規創業にIT担当取締役として参画し
新規事業の立ち上げにおける業務のIT化を実現。

2001年4月に現在の株式会社マジカジャパンを設立。
2003年に後の特別非営利活動法人Seasarファウンデーションの創設に参画。
2006年から2011年まで、国立大学法人琉球大学産官学連携推進機構の非常勤講師。
2011年9月に米国エークリッパー・インクを設立。

著書に「楽々ERDレッスン」「SQL書き方ドリル」など多数。

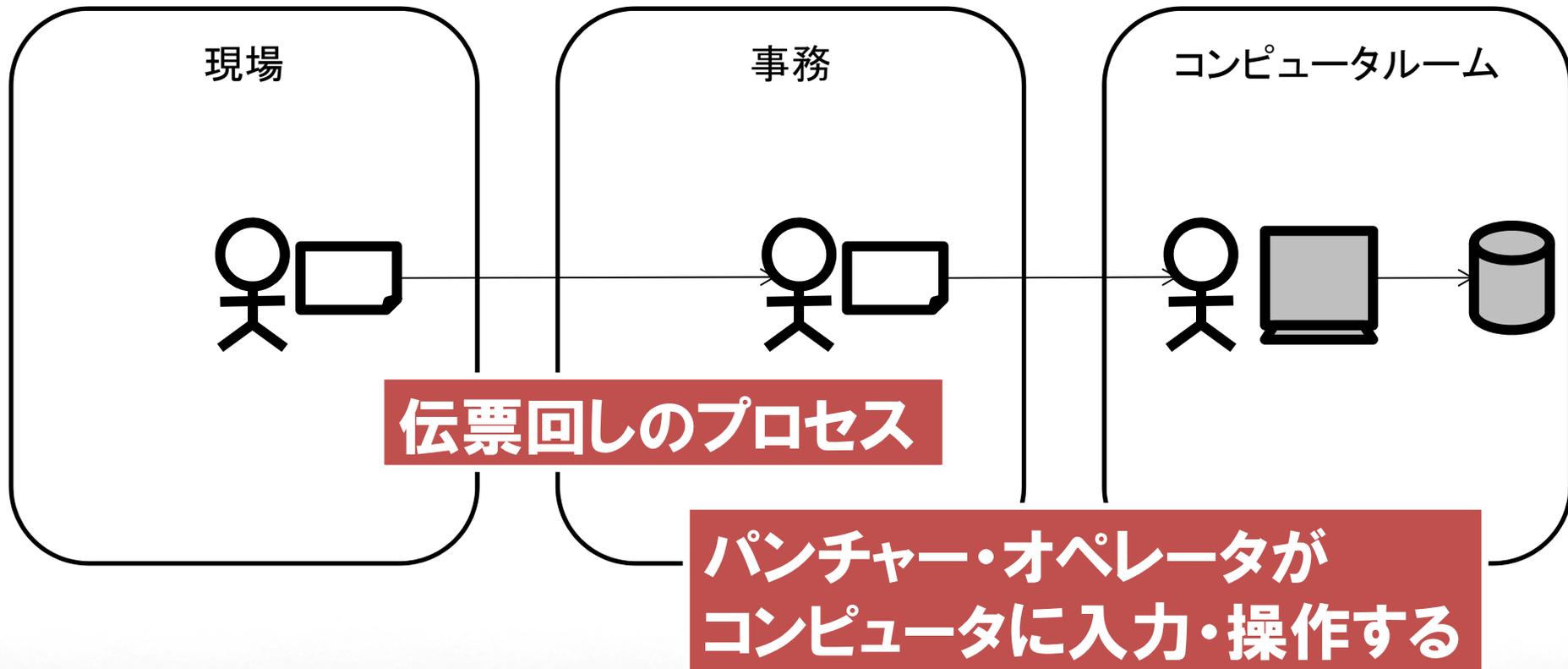




エンタープライズモバイルの ポテンシャルと課題

エンタープライズシステムの変遷

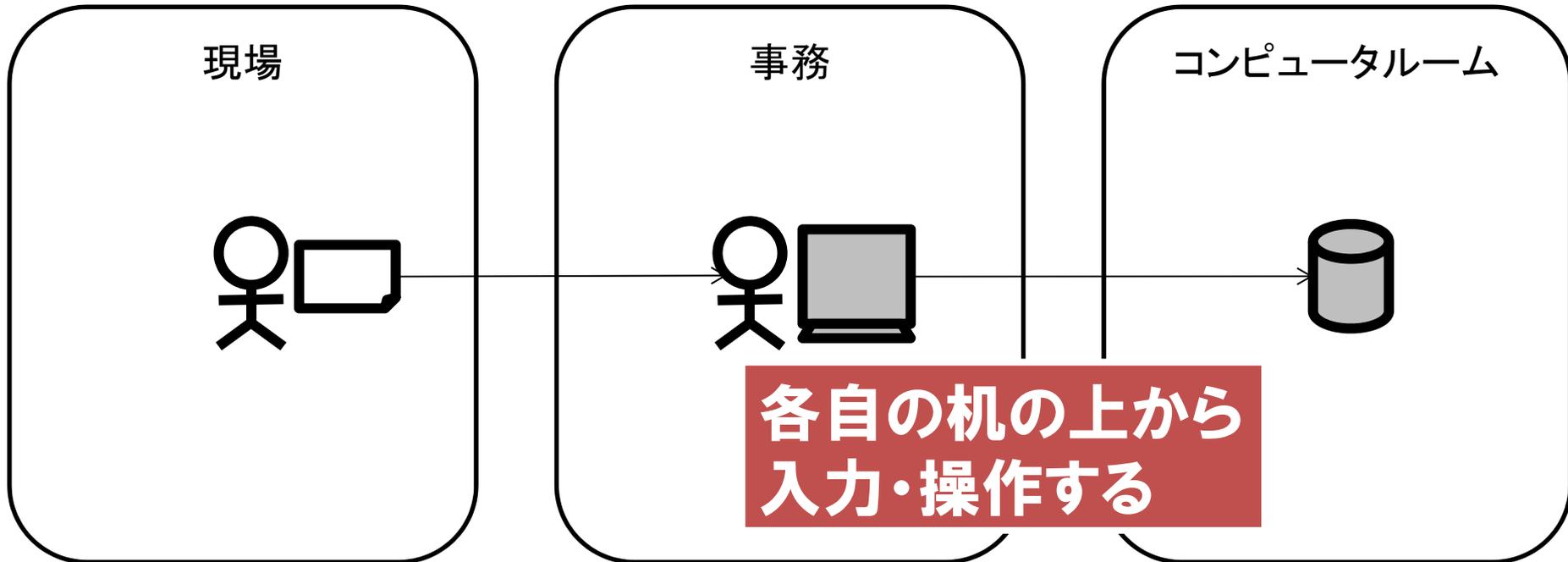
メインフレーム時代



エンタープライズシステムの変遷

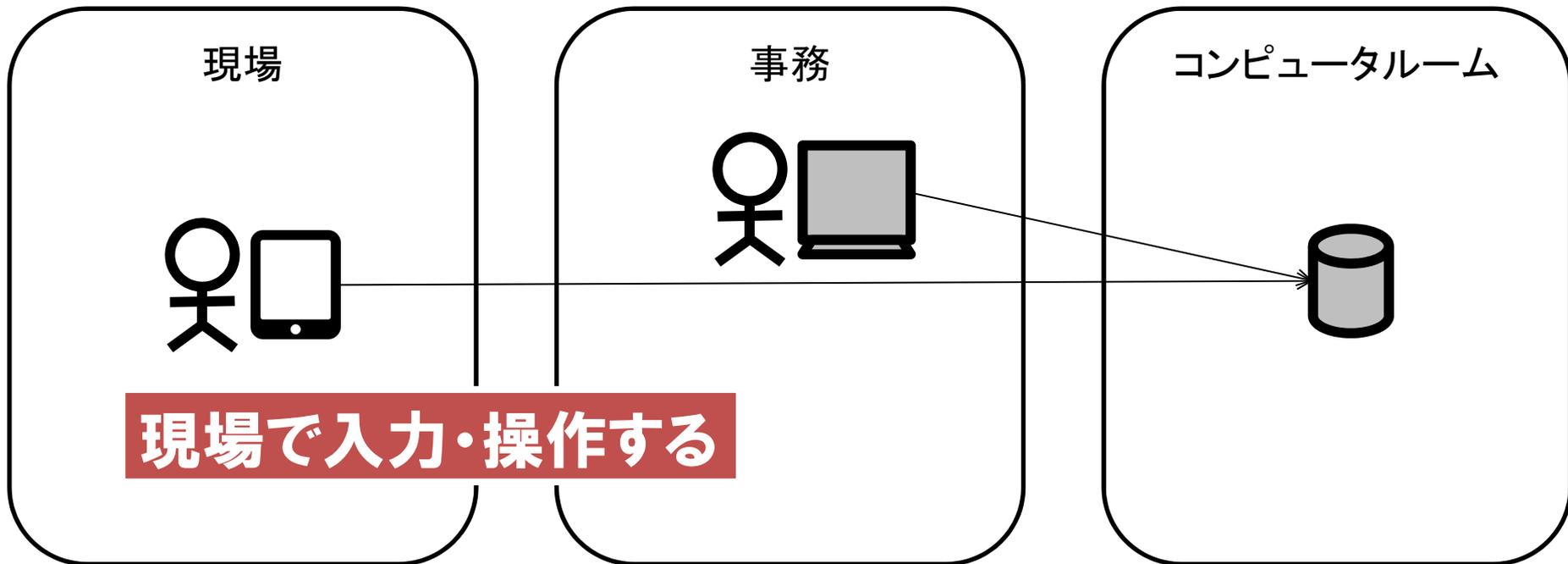
デスクトップPC時代

クラサバ、Web



エンタープライズシステムの変遷

モバイル時代

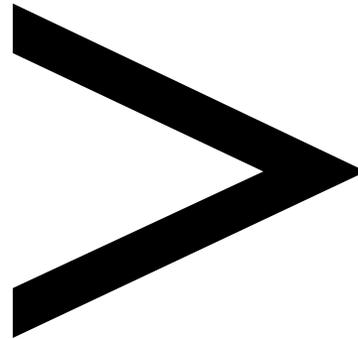


現場直結型 あるいは **すべてが現場になる**

手のひらの上に基幹系が乗る時代

最新のスマホ

20年前のメインフレーム

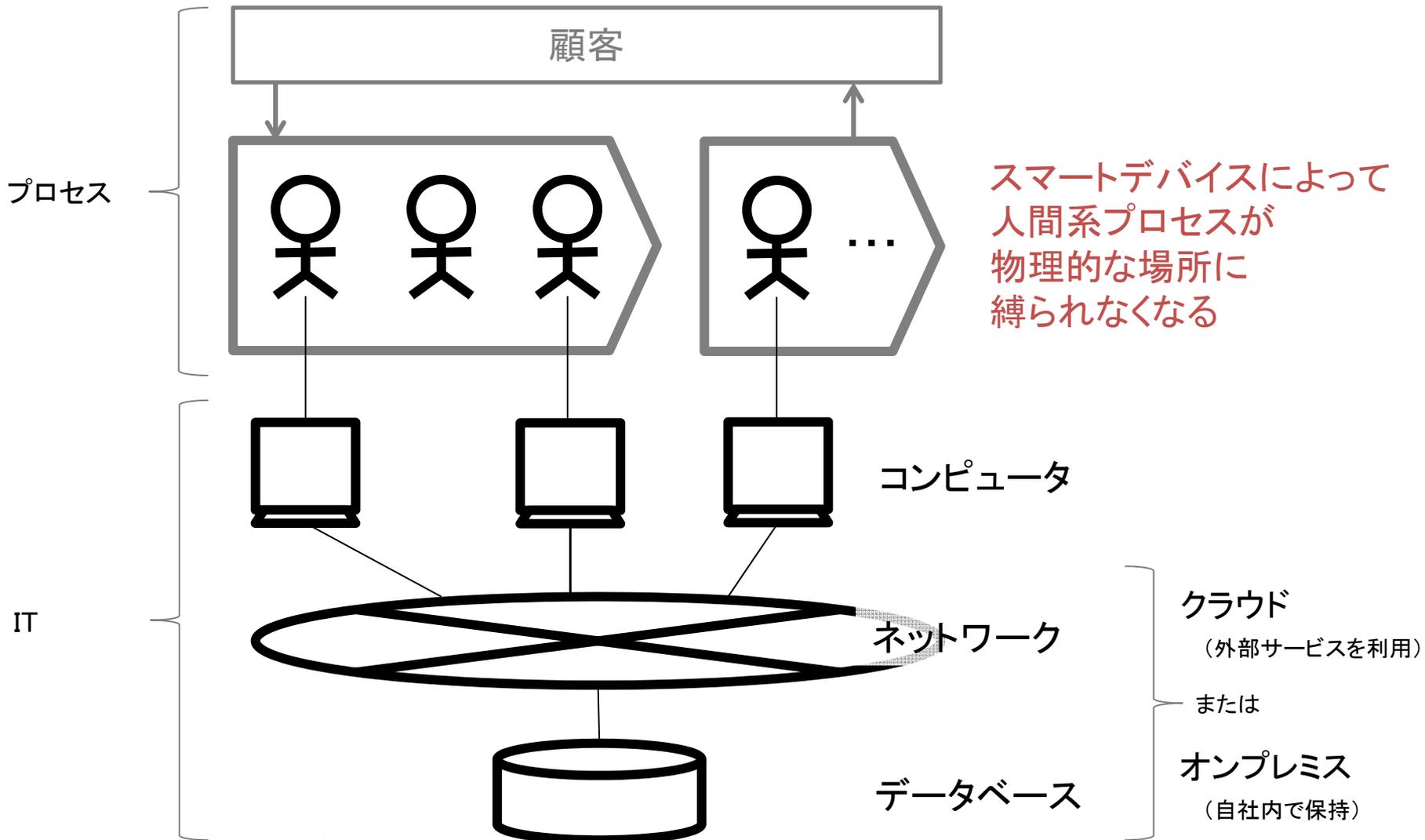


コンピュータに人間系プロセスを 合わせる時代の終焉

現場直結型だからこそその価値

- そもそもITの本質は「**中抜き**」の実現
 - セルフサービス、ダイレクトサービスへ
 - 既実証済
 - ネット証券
 - ネットショッピング(Amazonとか)
 - パンチャーやオペレーターも既に中抜きされた
- デスクワークから「**ながらワーク**」へ
 - 移動しながら
 - 接客しながら
 - 修理しながら
 - etc...

プロセスとITの関係構造



エンタープライズにおけるUX

- UX＝ユーザエクスペリエンス
 - ユーザの体験
 - でも自社の社員の体験を高めても仕方ないよね？
 - 重要なのは仕事の実行力を高めること
 - 良い仕事を出来るようにすることが、結果として成功という最高の体験を実現することになる
- **エクスペリエンスではなくエグゼキューション！**

エンタープライズUX実現のために

- リプレースは意味が無い
 - 既存プロセスを維持したまま、道具だけモバイルにしても効果は出ない
 - 古いパラダイムのプロセスのままだから
 - 開発工数は以前より増大している
 - アーキテクチャが複雑化している
 - UIの表現力向上に伴う実装の手間も増えている
- まだやっていないことをやろう！
 - 中抜きのを活かす
 - 「ながらワーク」だから出来る価値を考える

スマートデバイスは一過性か

- 結論から言うと「Windows3.1以上の変革期」
 - Webはクラサバの応用でありデスクトップパラダイム
 - モバイルは久しぶりのパラダイムシフト
- 教育コストの観点
 - 最近の学生は卒論をスマホで書く
 - パートの人達もスマホ慣れしている
- 業務系で利益貢献する事例の増加
 - 特に飲食系の注文端末
 - 人材不足への対応
 - プロセスの中抜き

**今後のメインストリームは
モバイルだと断言できる**

スマートデバイスらしいUIの必要性

- スマートデバイスによって
 - 顧客をビジネスプロセスに組み込む
 - 現場担当者をリアルタイムでサポートする
- でもWindows向けのGUIをそのままブチ込むと？

**そこでゲームUIに学ぶ
必要性が生まれる**



あらためてUIということを 考える

何故ゲームUIなのか

- 実はエンタープライズにおけるUIの知見は殆ど存在しない
 - 技術制約に従って、無理やり・取り敢えず、押し込んでだけのUIが大半を占める
 - 何故そのUIなのか説明できるシステムはほぼ皆無
 - プラットフォームのガイドラインを知らないまま設計してるケースが多い
- 結果として前例主義・過去の焼き直しに陥っている
 - Windowsでメインフレーム時代のUI: ファンクションキーとか
 - HTMLでWindowsのGUIを再現するとか
- ぶっちゃけ業務系のUIはダサい
 - パッと見て分かり辛いものが多い
- 別の手本が必要

何故ゲームUIなのか

- コンピュータ用のUIとしての歴史
- 幅広いユーザへの対応
- 操作マニュアルを読まなくても済むようにする配慮
- コンシューマ向けという苛酷さ

コンシューマ向けUI

ゲーム



らくらくアプリ



SNS



目を細めてよく見ると

- 選択指向のUI＝選んで押すだけ
 - ゲームは典型的
 - スタンプもボタン
 - ソフトキーボードもボタンの集合
- 世界観の中での統一性・一貫性
 - 標準的なパーツにこだわってはいない
 - ゲームだとむしろつまらなくなる

選択肢を設計する

- いかにかTEditを排除するか
- 選択肢が多すぎると逆効果
 - 年月日とか
 - Pickerは本当に使いやすいのか？
- 選択肢のための、
いわゆるマスターテーブルが必要になる
 - これを面倒くさがると、
ユーザに手間を押し付けることになる

例として

担当者を選択します X

 羽生

 可世木

 佐藤

⋮

確定する

例として

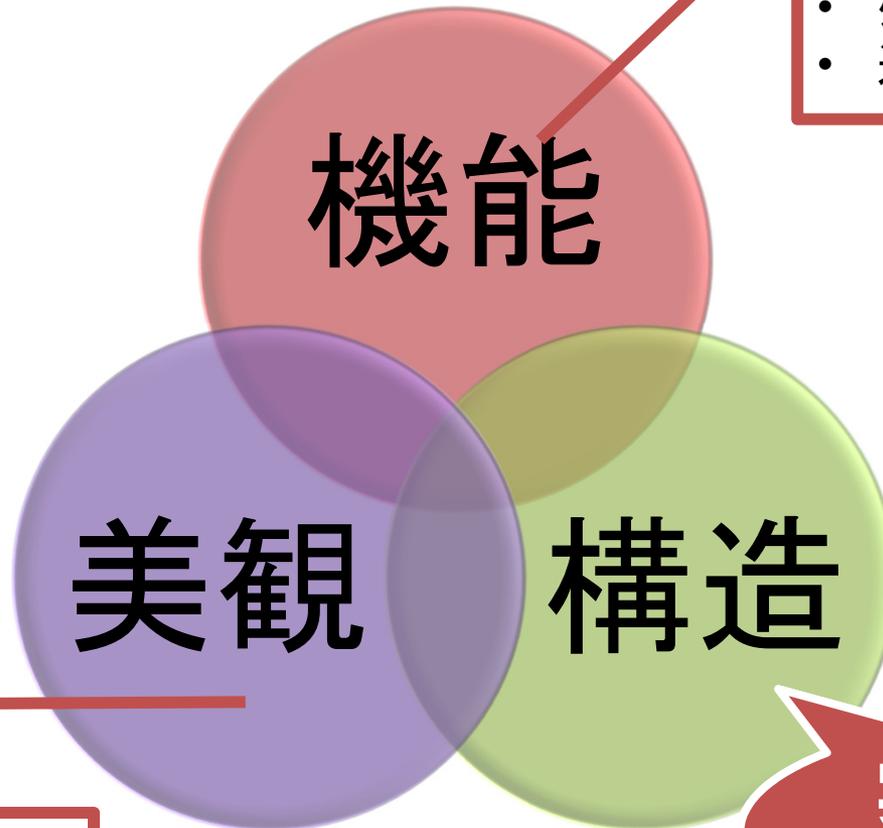
生年月日を入力します X

年 月 日

19	1	2	3	1	2	3	1	2	3
20	4	5	6	4	5	6	4	5	6
	7	8	9	7	8	9	7	8	9
	0			10	11	12			

確定する

UIデザインの三要素



- 処理や項目は機能
- 選択肢も機能

- カラーリング
- タイポグラフィ
- グラフィック

実はここが
課題

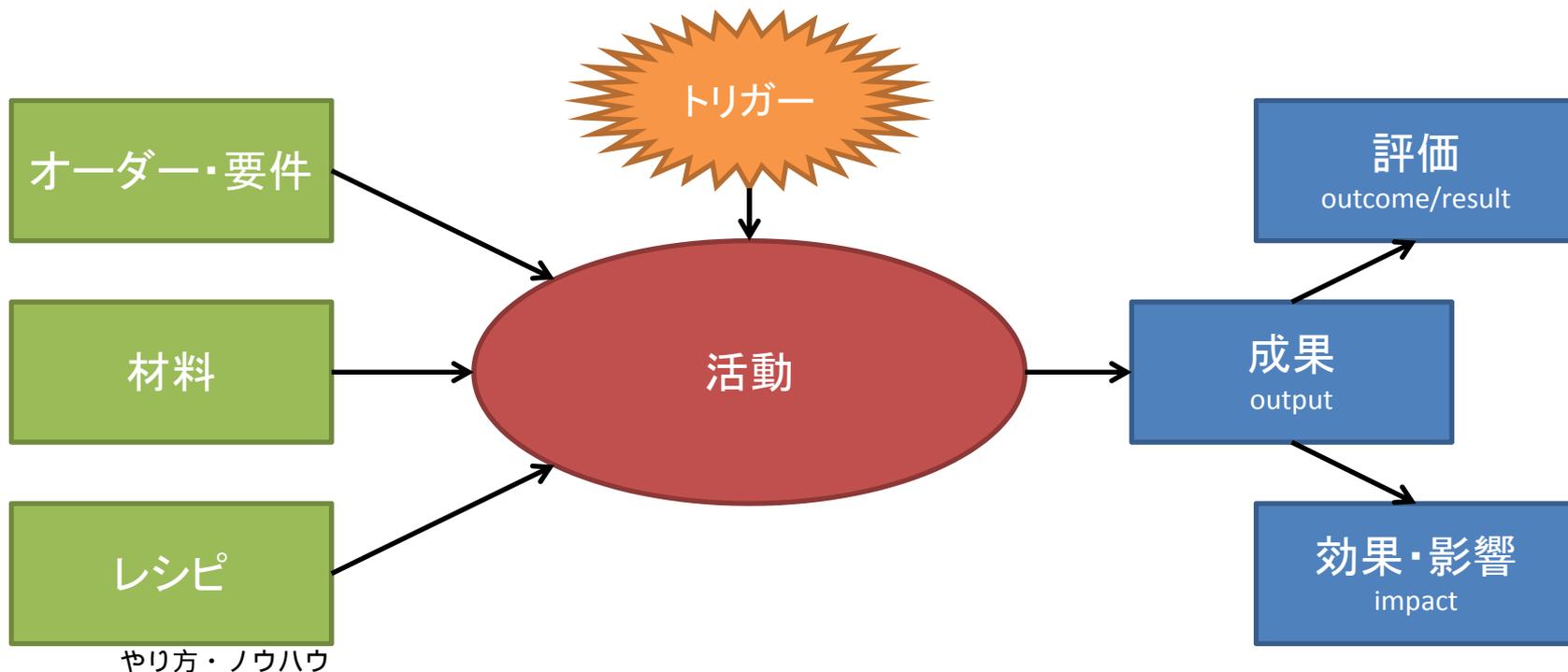
エンタープライズ特有の課題

- 項目数が多い！
- 小さな画面にいかに収めるかが課題
- UIの正規化・構造化ということ
 - 正規化: ノーマライゼーション
 - 正しいというよりもノーマル(普通)にするということ
- **レイアウトと画面遷移が肝！**

UIの構造化とスコープ定義

- レイアウトと画面遷移を適切に設計するには？
- アプリケーションごとの適切なスコープ定義が不可欠！
- ユーザのタスクごとにスコープを分離する
 - 複数のタスクを同一UIに分けない
- では、ユーザタスクをどう定義する？

ユーザタスクを考える



ユーザタスク=ユーザの仕事
成果が異なれば違う仕事である



FireMonkeyでのアプローチ

FireMonkeyに落としこむ

- デザインファースト vs コンポーネントファースト
 - いっぱい並んでるコンポーネントをどう使う？
- デザインを優先する！
 - 標準装備のコンポーネントに振り回されない！
 - 利用するコンポーネントを敢えて絞る
- スタイル重要！！

FMの学ぶべきポイント

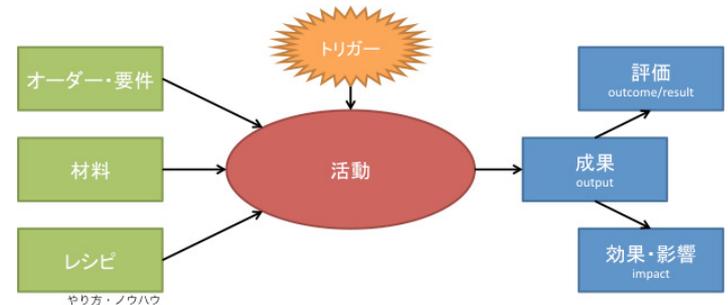
- スタイルの基本的な知識
- レイアウト系のコンポーネント
- レイアウト系のプロパティ
 - Align, Margin, Paddingなど
- フレーム
- データモジュール
- アクションリスト

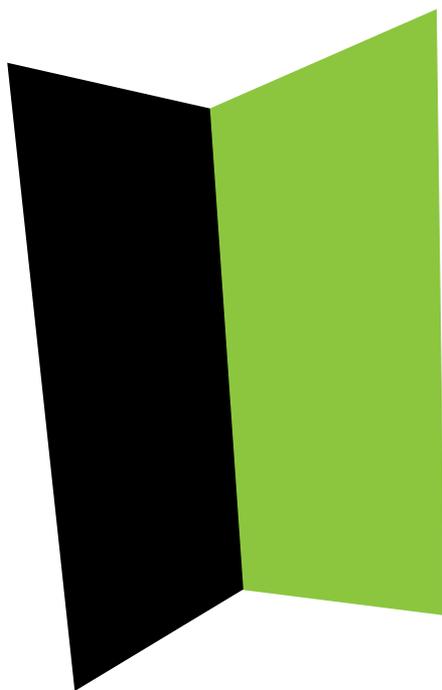
例えばFMでフラットデザイン風に

demo

いきなりIDEに向かうな！

- 仕様重要！！
- まずオーダーを固める！
 - 立ちながらUIを考える
 - 「ながらワーク」をイメージする
- そもそも何故そのアプリが必要なのか？
 - 実現したいプロセスの設計こそが実は肝！





まとめ

まとめ

- スマートデバイスは今後の主流
- 現場直結型時代の到来
- 現場の実行力向上がエンタープライズUX
- 現場の足を引っ張るUIじゃダメ
- UIに対するリテラシーをコンシューマ向けから学ぼう
- 適切なUIは適切なプロセス設計から
- FireMonkeyの既存コンポーネントに振り回されない
- 仕様重要！

ありがとう
ございました